



Universidade de Brasília - Darcy Ribeiro
Faculdade de Tecnologia - Departamento de Engenharia Elétrica

ENE0167 - Controle Digital - Turma A 2021/2

**Laboratório de Controle Digital:
Controle PI no espaço de estados da posição de um
servomecanismo linear**

Professor: Henrique Cezar Ferreira

Gabriel Tambara Rabelo
18/0017021
rabelo.gabriel@aluno.unb.br

Pedro Pereira Nunes
17/00020738
170020738@aluno.unb.br

4 de maio de 2022

Abstract

In order to control a linearly acting servomechanism, different system identification methods are compared after modeling the physical system. Through an approach that considers the system as being of 2nd order and with modeling in state space, a proportional and integrative controller was developed via state feedback with observer to achieve the desired dynamic responses of the system. All calculations and simulations were developed using Matlab software and complete control validation was performed by Simulink.

Keywords: Control. Digital. Servomechanism. Linear. State Space. PI Controller.

Resumo

Visando controlar um servomecanismo de atuação linear, diferentes métodos de identificação de sistema são comparados após a modelagem do sistema físico. Através de uma aproximação que considera o sistema como sendo de 2ª ordem e com modelagem em espaço de estados, foi desenvolvido um controlador proporcional e integrativo via realimentação de estados com observador para alcançar respostas dinâmicas do sistema desejadas. Todos os cálculos e simulações foram desenvolvidos por meio do software Matlab e a validação completa do controle foi realizada pelo Simulink.

Palavras-chave: Controle. Digital. Servomotor. Linear. Espaço de Estados. Controlador PI.

Conteúdo

1	Objetivo	4
2	Introdução	4
2.1	Equipamentos	4
2.2	Servomotor	5
2.3	Modelagem física	5
2.4	Sistemas de segunda ordem	6
2.5	Espaço de estados	7
2.6	Realimentação de estados	7
2.7	Observador de estados	8
2.8	Controle de servomotores no espaço de estados	9
3	Desenvolvimento	10
3.1	Modelagem do sistema	10
3.2	Identificação de parâmetros	11
3.2.1	Identificação dos parâmetros em Malha Aberta	12
3.3	Realização	12
3.4	Discretização	13
3.5	Características do Sistema	13
3.6	Observador	14
3.7	Calculo das matrizes K_1 e K_2	14
3.8	Avaliação Teórica do Sistema	14
3.8.1	Resposta à perturbação	15
3.8.2	Resposta a outras entradas	16
3.8.3	Mudança de Parâmetros da Planta	16
3.9	Avaliação prática do sistema	17
4	Conclusão	18
5	Referências	18
6	Anexos	19
6.1	Imagens grandes	19
6.2	Código Matlab	22

1 Objetivo

Desenvolvimento da proposta de experimento cujo intuito é agregar à formação dos estudantes na área de controle digital ao projetar um controlador discretizado de um sistema modelado no espaço de estados representando um servomotor de atuação linear. O projeto envolve a identificação do sistema baseado em dados obtidos experimentalmente e o projeto de um controlador PI por realimentação de estados com observador. Objetiva-se que o aluno seja capaz de compreender e interpretar os conceitos chave utilizados, bem como desenvolver habilidades e noções relacionadas ao tipo de projeto.

2 Introdução

2.1 Equipamentos

O ambiente de trabalho, no qual o projeto de controle a ser apresentado foi desenvolvido, é composto principalmente dos seguintes recursos presentes no laboratório de controle da Universidade de Brasília:

- Módulo de potência Quanser VoltPAQ;
- Placa de aquisição de dados Quanser Q8-USB;
- Planta servo linear Quanser IP02;
- Programa de controle em tempo real: QUARC-Simulink.

Neste cenário, a planta, representada pela Fig.[1], é responsável pelo papel de atuador do sistema, cuja movimentação deverá ser controlada pelo sistema a ser projetado e processado pelo bloco de controle em tempo real Quarc-Simulink. Esse bloco auxiliará no design, desenvolvimento, deploy e validação da aplicação carregada pelo simulink, que será o software de implementação desse mesmo modelo. Quanto a especificações técnicas, obtêm-se do manual da Quanser que a tensão nominal de operação do motor é de aproximadamente 6 Volts.

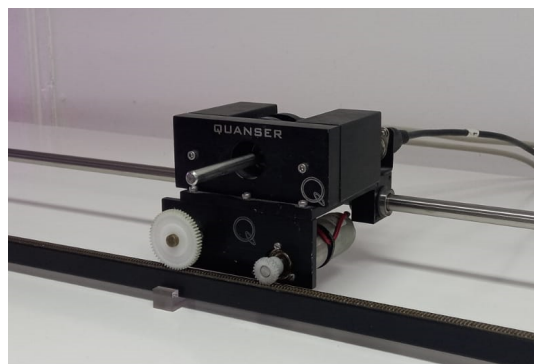


Figura 1: Planta servo linear Quanser IP02

O VoltPAQ é o amplificador controlado por tensão ideal para configurações complexas de sistemas em tempo real realimentados com finalidade educacional e pesquisa, de modo a permitir o adequado controle do sistema. Por fim, a placa de aquisição de dados Quanser Q8-USB é um dispositivo de aquisição de dados de oito canais de E/S de ponto único que oferece desempenho confiável em tempo real por meio de uma interface USB, podendo, portanto, se comunicar adequadamente com o computador e o sistema. Esses dois componentes podem ser observados fisicamente pela Fig.[2].



Figura 2: Módulo de potência e placa de aquisição de dados Quanser

2.2 Servomotor

Servomotores são motores recorrentemente aplicados em sistemas de controle de precisão e maquinário industrial, muitas vezes utilizados em sistemas de malha fechada, tendo em vista que são acompanhados por encoders ou outros dispositivos de medição. São utilizados para obtenção de uma resposta prática do sistema dada uma entrada ideal.

São diferentes de motores de passo, que realizam o movimento de forma discretizada pelo número de passos a serem seguidos, logo não é necessário o uso de coleta de dados por realimentação de forma a garantir o movimento desejado.

Outra característica importante dos motores é que possuem Zona Morta, característica responsável por alterar o funcionamento do seu acionamento dada condição inicial. Esse efeito por vezes é percebido como um atraso na entrada, visto a excitação inicial não ocasionar movimentação imediata, portanto este efeito é enxergado como um atraso no sistema, que pode ou não ser desprezível.

2.3 Modelagem física

Objetivando encontrar o modelo dinâmico da planta servo linear Quanser, se faz necessária a análise da sua composição. De forma simplificada ele possui a seguinte topologia, representada pelo diagrama da Fig.[3].

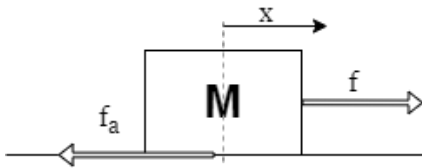


Figura 3: Diagrama de blocos do sistema

Sendo $f(t)$ a força imposta no sistema de massa M pelo motor, $x(t)$ o deslocamento do atuador e f_a a força de atrito viscoso. Portanto a dinâmica do sistema pode ser descrita pela Eq.[1].

$$M \frac{d^2 x}{dt^2} = f - K_{at} \frac{dx}{dt} \quad (1)$$

Considerando a força de atrito uma proporção da velocidade pela constante de atrito K_{at} .

O intuito é relacionar a posição, a variável a ser controlada, com a ação de atuação, que se dará pela tensão aplicada ao motor, V_m . Portanto, para a modelagem do circuito do motor, o modelo simplificado é representado pelo diagrama da Fig.[4].

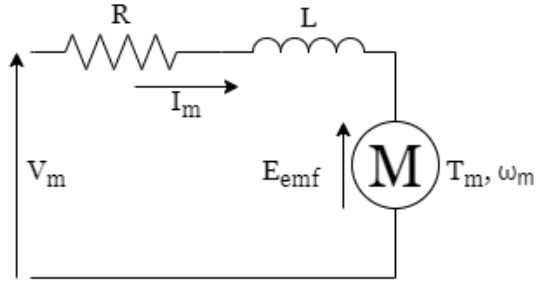


Figura 4: Diagrama do circuito da planta servo

Sendo definidas E_{emf} , a força eletromotriz contrária do motor, de valor igual ao produto da constante K_m pela rotação. O torque é definido como T_m e a rotação como ω_m . Também são definidas a resistência R e a indutância L .

Fazendo a análise do circuito, pode-se encontrar a relação dada pela Eq.[2], que denota a tensão em função dos parâmetros de modelo.

$$V_m = E_{emf} + L \frac{dI_m}{dx} + RI_m \quad (2)$$

Pode se relacionar o torque de rotação do motor com a corrente I_m , com o rendimento η_m e pela constante de torque K_t , como definido pela Eq.[3].

$$T_m = \eta_m K_t I_m \quad (3)$$

Pode-se representar o torque do motor pela força F_c que o mesmo faz através do pinhão, de raio r , que serve como braço de alavanca. Também são considerados a eficiência do sistema de engrenagens η_g e sua razão de transmissão K_g , como é demonstrado pela Eq.[4].

$$F_c = \frac{\eta_g K_g T_m}{r} \quad (4)$$

Por fim, relacionando a rotação do motor com o deslocamento do sistema, é válida a Eq.[5].

$$\omega_m = \frac{K_g}{r} \frac{dx}{dt} \quad (5)$$

Como é usual nesse tipo de cenário que a indutância seja muito menor que a resistência, é feita uma aproximação do modelo ao se desprezar o efeito da indutância. Outra adaptação é a substituição da força contra eletromotriz pela sua proporção em \mathbf{K}_m , da velocidade ω_m .

Substituindo as equações resultantes na Eq.[1] e realizando a transformada de Laplace, é obtida a seguinte função de transferência, apresentada pela Eq.[6].

$$\frac{X(s)}{V_m(s)} = \frac{\frac{V_m r \eta_g \eta_m K_g K_t}{R_m r^2}}{s \left(M s + \left(\frac{\eta_g \eta_m K_g K_t K_m}{R_m r^2} \right) \right)} \quad (6)$$

Que pode ser simplificada para o formato dado pela Eq.[7].

$$\frac{X(s)}{V_m(s)} = \frac{K}{s(1 + \tau s)} \quad (7)$$

2.4 Sistemas de segunda ordem

Sistemas de controle podem ser representados de diversas formas. Dentre as mais úteis são os modelos no domínio da frequência, em específico, o sistema de segunda ordem cujo formato é apresentado pela Eq.[8].

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8)$$

Nessa equação, pode-se observar a frequência natural do sistema ω_n , dada em radianos por segundo e que representa a frequência na qual um sistema tende a oscilar quando na ausência de interferências, o fator de amortecimento ζ , que representa o quanto o sistema tende a oscilar antes de alcançar o estado estacionário.

Sistemas de segunda ordem podem ser categorizados com base em seu valor da constante de amortecimento com diferentes propriedades oscilatórias, onde

têm-se que quando é zero o sistema é dito não amortecido, entre zero e um é dito subamortecido, se igual a um é dito criticamente amortecido e se maior a um é dito superamortecido.

Algumas das características dos sistemas de segunda ordem são o instante de pico, que representa o tempo desde o início do pulso de sinal até o instante onde o maior valor é atingido no sistema controlado. O seu valor é dado pela Eq.[9].

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (9)$$

Outra característica de interesse é o máximo sobressinal, que representa o quanto do sinal de convergência do sistema é obtido durante o maior pico, ou seja, no instante de pico, e seu valor é dado pela Eq.[10].

$$M_p = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (10)$$

Por fim, sistemas de segunda ordem também são caracterizados por seu tempo de acomodação, que envolve o tempo necessário para que a resposta do sistema se estabilize e não passe de uma certa porcentagem do seu valor final. Para a porcentagem usual de 2%, a equação que a define é dada pela Eq.[11].

$$t_s = \frac{4}{\omega_n \zeta} \quad (11)$$

Para o equivalente do sistema amostrado, variando em s contínuo, ou z discreto, com período de amostragem T , os polos desse sistema são dados pela Eq.[12] no domínio contínuo e pela Eq.[13] no domínio discreto, em função apenas dessas constantes apresentadas.

$$s = \omega_n \angle \cos^{-1}(\zeta) \quad (12)$$

$$z = e^{-\sigma T} \angle \omega_n T \sqrt{1 - \zeta^2} \quad (13)$$

$$\sigma = \zeta \omega_n$$

Essas equações e suas respectivas deduções podem ser obtidas em [2].

Sistemas de segunda ordem usualmente possuem uma adequada estabilidade para 8 a 10 amostras por

ciclo de oscilação, vide [p. 216][1]. Pode-se obter a quantidade de amostrar de um sistema dadas as suas propriedades dinâmicas, vide [p. 221][1], sendo o valor igual à frequência natural dividido pela frequência amortecida, como dado pela Eq.[14]

$$n = \frac{2\pi}{\omega_n T \sqrt{1 - \zeta^2}} \quad (14)$$

2.5 Espaço de estados

Trabalhar com múltiplos sistemas de controle, ou sistemas com múltiplas entradas ou saídas pode ser exaustivo dado o escopo do projeto, portanto, se faz necessário o uso de uma nova representação, denominada espaço de estados, na qual o sistema é representado em função da suas entradas, saídas e variáveis de estado. A sua forma matemática é dada pelas Eqs.[15, , 16], na representação contínua.

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (15)$$

$$\mathbf{y} = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \quad (16)$$

Aqui definem-se as matrizes **A**, **B**, **C** e **D**, denominadas juntas realização do sistema, sendo **x** a variável de estado ou vetor de variáveis de estado e **u** a entrada do sistema ou vetor equivalente. É usual dentro do escopo tratado que em sistemas de controle digital considere-se que a matriz **D** seja nula, de modo que a saída não dependa diretamente da entrada. Ademais, pode-se visualizar o diagrama que representa esse sistema no espaço de estados na Fig.[5].

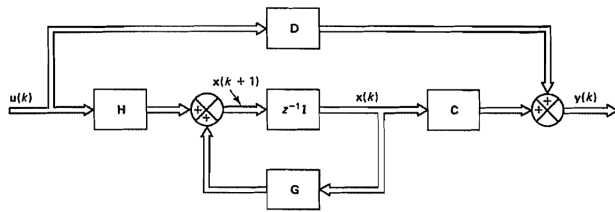


Figura 5: Diagrama sistema no espaço de estados [p. 296][1].

Nessa representação de espaço de estados, o sistema pode ser dividido em múltiplas equações diferenciais que quando representadas no domínio Z, que é o domínio de cálculo discretizado, possuem um valor diferente para as matrizes **A** e **B**, então renomeadas como **G** e **H**, cuja relação é dada por:

$$\mathbf{G}(kT) = \mathbf{G}[k] = e^{AT} \quad (17)$$

$$\mathbf{H}(kT) = \mathbf{H}[k] = \left(\int_0^T e^{A\lambda} d\lambda \right) \cdot B \quad (18)$$

Duas das propriedades mais importantes de um sistema na forma do espaço de estados são sua observabilidade e controlabilidade. A Observabilidade avalia a capacidade de um sistema de estimar os estados do mesmo utilizando da matriz de observabilidade, enquanto a controlabilidade avalia a capacidade de se alterar os estados de um sistema, que também pode ser medida pela matriz de controlabilidade. O cálculo para sua obtenção é apresentado pelas Eqs[19, 20], e o seu equivalente para o domínio discretizado também é válido.

$$\mathcal{O}_v = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (19)$$

$$\mathbf{C}_v = [\mathbf{B} \quad \mathbf{AB} \quad \dots \quad \mathbf{AB}^{n-1}] \quad (20)$$

2.6 Realimentação de estados

Por vezes se faz necessário que um sistema possua certas características dinâmicas como tempo de subida, tempo de pico, máximo sobressinal, entre outros. As propriedades que diretamente se relacionam com essas características são os polos de malha fechada do sistema, que podem ser também representados pelos autovalores da matriz **A**, ou **G** no domínio discretizado. Caso os autovalores não possuam os valores de interesse, se faz necessário uma mudança nos mesmos para atribuir essas características, e isso é feito através da realimentação de estados. Um requisito essencial para que a realimentação seja imple-

mentada em um dado sistema é que esse sistema seja dito controlável.

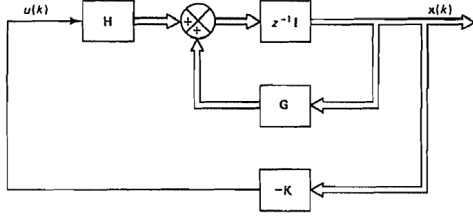


Figura 6: Diagrama de sistema com realimentação de estados [p. 403][1].

A Fig.[6]. representa uma realimentação de estados cuja matriz de realimentação é \mathbf{K} . Caso o interesse seja definir os autovalores como $\mu_1, \mu_2, \dots, \mu_n$, basta projetar um valor para \mathbf{K} que satisfaça a Eq.[21], sabendo que os autovalores da matriz que multiplica o vetor de estados são os polos de malha fechada.

$$\mathbf{x}[k+1] = (\mathbf{A} - \mathbf{H}\mathbf{K})\mathbf{x}[k] \quad (21)$$

2.7 Observador de estados

Em sistemas de controle, é comum se deparar com variáveis de estado que não podem ser diretamente observadas. Por exemplo, é usual que apenas as variáveis de saída de um sistema sejam observadas. Essa situação é contornada pela estimação desses estados pelo que se chama observador através das variáveis que se tem acesso, as de controle e saída. Uma condição necessária para a implementação de um observador é a observabilidade do sistema.

Define-se $\tilde{\mathbf{x}}$ como o estado estimado pelo observador a partir de \mathbf{y} e \mathbf{u} . Portanto, segue que a saída do sistema estimada segue o formato representado pela Eq.[22].

$$\tilde{\mathbf{y}}[k] = \mathbf{C}\tilde{\mathbf{x}}[k] \quad (22)$$

A seguir, para melhorar o desempenho do projeto, utiliza-se a diferença entre a saída real e a saída estimada por um ganho \mathbf{K}_e de modo que a equação de estados toma a forma da Eq.[23].

$$\tilde{\mathbf{x}}[k+1] = \mathbf{G}\tilde{\mathbf{x}}[k] + \mathbf{H}\mathbf{u}[k] + \mathbf{K}_e[\mathbf{y}[k] - \mathbf{C}\tilde{\mathbf{x}}[k]] \quad (23)$$

Considere agora o caso de uma das variáveis de estado não estar disponível para observação. Para que o sistema do observador de estados possa estimar esse estado e para que ele mantenha as mesmas matrizes \mathbf{G} e \mathbf{H} nas equações que descrevem o espaço de estados de forma a melhorar a resposta, uma melhoria na topologia é feita sob um sistema com realimentação de estados, com a matriz de ganho de realimentação do observador, \mathbf{K} , tomando a realimentação pela ação de controle dada pela Eq.[24], e recebe o nome de observador de estados de ordem completa, a qual é representada na Fig.[7].

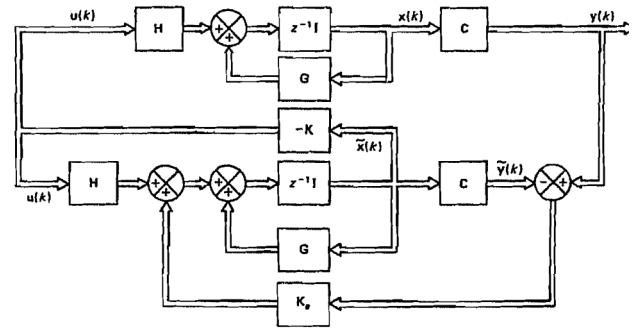


Figura 7: Diagrama do observador de estados de ordem completa [p. 428][1].

$$\mathbf{u}[k] = -\mathbf{K}\tilde{\mathbf{x}}[k] \quad (24)$$

E pode-se algebricamente chegar à adaptada equação de estados estimados como representado pela Fig.[25]:

$$\tilde{\mathbf{x}}[k+1] = (\mathbf{G} - \mathbf{K}_e\mathbf{C})\tilde{\mathbf{x}}[k] + \mathbf{H}\mathbf{u}[k] + \mathbf{K}_e\mathbf{y}[k] \quad (25)$$

A fim de se obter um adequado observador de estados, deve-se definir polos de observação através da matriz \mathbf{K}_e . O recomendado pela teoria é que os polos do observador sejam ao menos 5 vezes menores, ou mais rápidos, do que os polos do sistema, de forma

a garantir uma boa resposta. Como a topologia do observador é análoga à da realimentação, pode-se definir o valor de \mathbf{K}_e pelos mesmos métodos que a matriz de realimentação, como o método de Ackermann, vide [4].

2.8 Controle de servomotores no espaço de estados

Para realizar o controle de um sistema que compõe principalmente um servomotor, é de interesse remover o erro no estado estacionário para entradas do tipo degrau, que são comuns em sua atuação. Para isso, é necessário a inserção de um ou mais integradores no sistema. Além disso, o controle do servomotor envolverá uma realimentação de estados, cuja finalidade é definir as propriedades do sistema como tempo de pico, tempo de acomodação e máximo sobressinal, através da alocação de polos, além do componente integrador, e se necessário, um observador.

O formato do sistema com as implementações toma a forma representada pela Fig.[8].

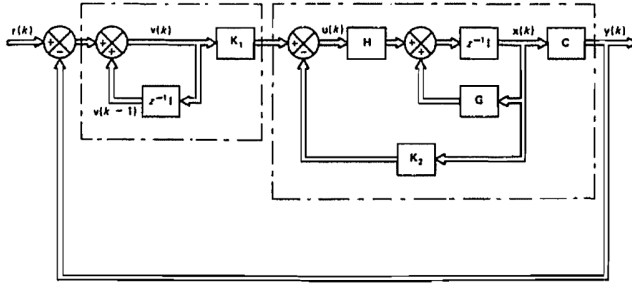


Figura 8: Diagrama de sistema controlado no espaço de estados [p. 461][1].

Quanto à realimentação, representada pelo sub-diagrama à direita na Fig.[8], percebe-se que com essa nova estrutura o sistema possui uma matriz \mathbf{K}_2 capaz de afetar o polinômio característico do sistema, e consequentemente os polos que definem as propriedades do sistema.

Uma das maneiras em que se é possível inserir um efeito integrativo no sistema é ao inserir um vetor

de estado $v[k]$ que integra a diferença de sinal entre a referência e a saída e é representado como o sub-diagrama à esquerda na Fig.[8]. Nesse caso, o estado integrativo possui uma matriz de ganho \mathbf{K}_1 que adiciona ao sistema mais um estado, que consequentemente aumenta o número de autovalores a serem calculados. Entretanto, como o sistema físico ainda é de segunda ordem, deseja-se que o pólo do integrador seja quão rápido quanto for possível.

Analisando todo o sistema, obtêm-se a seguinte nova relação:

$$v[k+1] = v[k] + r[k+1] - y[k+1] \quad (26)$$

Pela análise algébrica relacionando as variáveis do sistema, têm-se:

$$v[k+1] = -\mathbf{CGx}[k] + v[k] - \mathbf{CHu}[k] + r[k+1] \quad (27)$$

Observa-se que o vetor $\mathbf{u}[k]$ é uma combinação linear dos estados $\mathbf{x}[k]$ e $\mathbf{v}[k]$ do sistema pelas matrizes \mathbf{K}_1 e \mathbf{K}_2 , de modo que pode-se representar o novo vetor de estados em função de $\mathbf{x}[k]$ e $\mathbf{u}[k]$. Seguindo diretamente dessa proposição da Eq.[27] e do diagrama do sistema representado pela Fig.[8], obtêm-se o seguinte sistema:

$$\begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{u}[k+1] \end{bmatrix} = \mathbf{G}_n \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{u}[k] \end{bmatrix} + \mathbf{H}_n r[k+1] \quad (28)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{u}[k] \end{bmatrix} \quad (29)$$

$$\mathbf{G}_n = \begin{bmatrix} \mathbf{G} & \mathbf{H} \\ \mathbf{K}_2 - \mathbf{K}_2\mathbf{G} - \mathbf{K}_1\mathbf{CG} & \mathbf{I} - \mathbf{K}_2\mathbf{H} - \mathbf{K}_1\mathbf{CH} \end{bmatrix}$$

$$\mathbf{H}_n = \begin{bmatrix} 0 \\ \mathbf{K}_1 \end{bmatrix}$$

Pode-se observar que os polos do sistema em malha fechada, dados pelos autovalores da matriz \mathbf{G}_n podem ser definidos para um dado valor constante de $r[k]$, o que ocorre na entrada degrau. Também pode-se observar que no instante $k \rightarrow \infty$, para a Eq.[26],

o valor da saída toma a forma da referência, comprovando o erro nulo no estado estacionário.

Para se definir os valores das matrizes \mathbf{K}_1 e \mathbf{K}_2 , define-se um novo vetor auxiliar de estados ξ , dado pela diferença dos valores dos estados anteriores no instante k pelos seus valores no infinito, de forma que o termo referente à entrada é cancelado por ser constante. Em seguida, a matriz é modificada de modo que a sua segunda linha seja formada apenas por zeros, denotada $\hat{\mathbf{G}}$ em seu lugar, e uma nova matriz $\hat{\mathbf{H}}$ formada por um vetor coluna que multiplica uma nova variável de estado $\mathbf{w}[k]$, que juntos compensam os elementos removidos de \mathbf{G} . Esse sistema resultante ainda possui a matriz que acompanha o vetor de estados como a que define o comportamento dinâmico do sistema e suas equações resultantes têm a forma:

$$\xi[k+1] = \hat{\mathbf{G}}\xi[k] + \hat{\mathbf{H}}\mathbf{w}[k] \quad (30)$$

$$\mathbf{w}[k] = -\hat{\mathbf{K}}\xi[k] \quad (31)$$

$$\hat{\mathbf{G}} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \\ 0 & 0 \end{bmatrix} \quad \hat{\mathbf{H}} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \quad (32)$$

Assim, é possível observar que a matriz $\hat{\mathbf{K}}$, junto a matriz $\hat{\mathbf{G}}$, levam o sistema para os pólos desejados. De tal maneira, como a matriz $\hat{\mathbf{G}}$ é dada pela modelagem do sistema, basta calcular a matriz $\hat{\mathbf{K}}$ por algum método desejado, como a fórmula de Ackermann.

Utilizando de certa manipulação matricial, pode-se redefinir $\hat{\mathbf{K}}$ como se segue pela Eq.[33].

$$\hat{\mathbf{K}} = -[\mathbf{K}_2 - \mathbf{K}_2\mathbf{G} - \mathbf{K}_1\mathbf{C}\mathbf{G} \quad \mathbf{I} - \mathbf{K}_2\mathbf{H} - \mathbf{K}_1\mathbf{C}\mathbf{H}]$$

$$\hat{\mathbf{K}} = [\mathbf{K}_2 \quad \mathbf{K}_1] \begin{bmatrix} \mathbf{G} - \mathbf{I} & \mathbf{H} \\ \mathbf{C}\mathbf{G} & \mathbf{C}\mathbf{H} \end{bmatrix} + [0 \quad \mathbf{I}] \quad (33)$$

É de conhecimento teórico que se o sistema com \mathbf{G} e \mathbf{H} é controlável, o equivalente em $\hat{\mathbf{G}}$ e $\hat{\mathbf{H}}$ também o será. Portanto, pode-se utilizar do método usual de alocação de polos com esse novo sistema, ao se definir os autovalores de interesse pela matriz $\hat{\mathbf{K}}$, e consequentemente obtendo matematicamente as constantes para as matrizes \mathbf{K}_1 e \mathbf{K}_2 ao isolá-las na Eq.[33].

Portanto, segue que o formato final da representação de espaço e estados toma a forma das Eqs.[34,35].

$$\begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{v}[k+1] \end{bmatrix} = \mathbf{G}_f \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{v}[k] \end{bmatrix} + \mathbf{H}_f \mathbf{r}[k+1] \quad (34)$$

$$\mathbf{y}[k] = [\mathbf{C} \quad 0] \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{v}[k] \end{bmatrix} + \mathbf{D}\mathbf{r}[k+1] \quad (35)$$

$$\mathbf{G}_f = \begin{bmatrix} \mathbf{G} - \mathbf{H}\mathbf{K}_2 & \mathbf{H}\mathbf{K}_1 \\ -\mathbf{C}\mathbf{G} + \mathbf{C}\mathbf{H}\mathbf{K}_2 & \mathbf{I} - \mathbf{C}\mathbf{H}\mathbf{K}_1 \end{bmatrix} \quad \mathbf{H}_f = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}$$

O desenvolvimento matemático que leva às equações desta seção podem ser encontrados na [sec. 6.7][1].

3 Desenvolvimento

3.1 Modelagem do sistema

O sistema em questão é um servomecanismo cuja entrada é um sinal de referência posicional para o trilho conectado ao motor, e a saída é a localização efetiva do trilho no sistema. O motor fica posicionado na distância de 15cm, com o trilho tendo aproximadamente 30cm de comprimento. O sistema é comandado através do QUARC Real-Time Control Software, que possui interface junto ao Simulink-Matlab.

Para aquisição inicial dos dados para a identificação do sistema, o bloco QUARC, que comanda a tensão a ser enviada ao motor e retorna o deslocamento do motor, foi inserido em uma malha fechada com controle proporcional de ganho 420. Assim, o sistema foi excitado com uma entrada degrau de intensidade 15 e o motor se moveu de sua posição até alcançar aproximadamente o ponto 15 do deslocamento do motor. Com base nesse evento, pôde-se obter uma tabela que relaciona os dados de entrada, o tempo, e a saída, que representa a posição do servomecanismo no sistema.

Com essa tabela, extrai-se os vetores de tempo, de entrada e de saída, de modo que pode-se obter o maior valor de saída, e o seu instante de tempo respectivo. Com esses dados, é possível calcular os

parâmetros ω e ζ da Eq.[8]. Assim, por inspeção do formato de onda gerado pelos dados obtidos, e confirmado pelos resultados posteriores, discutidos na próxima seção, conclui-se que a aproximação matemática do sistema para um modelo segunda ordem é boa o suficiente para a nossa análise. A tabela gerada está no vetor de dados $\mathbf{x}()$ disponibilizado para o código do Matlab.

A identificação é feita com os dados obtidos do sistema em malha fechada com o controlador proporcional, para buscar algo semelhante à Eq.[8]. Entretanto como o sistema real não está por definição em malha fechada, mas sim em malha aberta, como representado pela Eq.[7], a equação que caracteriza o sistema se aproxima de:

$$\frac{Y(s)}{R(s)} = \frac{b}{s^2 + as} \quad (36)$$

Em que:

- $b = \omega^2 / \text{Ganho Proporcional}$
- $a = 2\zeta\omega$

3.2 Identificação de parâmetros

Com a Eq.[10] e o valor máximo conhecido, além do valor em estado estacionário, definido como 15 dada a entrada, pôde-se calcular a porcentagem do sobressinal e utilizar a fórmula para se encontrar o fator de amortecimento do sistema, de forma que $\zeta = 0.24352$. Utilizando agora a Eq.[9], pode-se utilizar o fator de amortecimento e o tempo de pico coletado experimentalmente para estimar o valor da frequência natural do sistema, de modo que $\omega_n = 20.24438$. Percebeu-se pelos dados obtidos que há um atraso de 0.033 segundos, de forma que sua representação na função também é relevante. Com essas duas variáveis estimadas, o sistema em malha fechada é descrito por:

$$\frac{Y(s)}{R(s)} = \frac{409.83}{s^2 + 9.86s + 409.83} e^{-0.033s} \quad (37)$$

Utilizando de ferramental disponível pela ferramenta Matlab, pôde-se simular o comportamento do modelo estimado e compará-lo graficamente com o

obtido experimentalmente pela tabela inicial de dados. O gráfico obtido é representado pela Fig.[9].

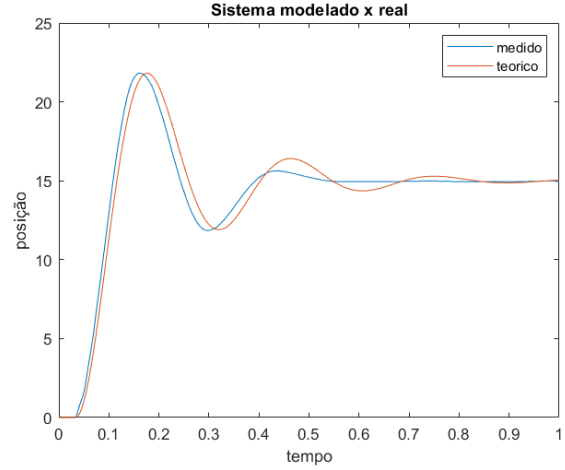


Figura 9: Modelo medido x estimado [cm x seg]

Pode-se perceber pelo gráfico que o modelo estimado ainda não se aproxima de forma satisfatória à forma de onda medida, de modo que uma alteração nos parâmetros sob método heurístico e iterado nos leva aos valores otimizados de $\omega_n = 24.5$ e $\zeta = 0.253$, cuja forma de onda é comparada e representada pela Fig.[10].

O novo, e adequado modelo, possui a seguinte equação, aproximada pelo Matlab:

$$\frac{Y(s)}{R(s)} = \frac{600.3}{s^2 + 12.4s + 600.3} e^{-0.033s} \quad (38)$$

Para fins de comparação, foi utilizada a ferramenta de identificação de sistemas do Matlab, *ident*, através da função *tfest*. A função *tfest* utiliza por padrão o método de identificação de sistemas conhecido como Variáveis Instrumentais (VI), que utiliza o método dos Mínimos Quadrados após a definição da matriz de entrada. O método *tfest* realiza a identificação de um sistema com a possibilidade de receber como argumento o atraso, número de pólos e zeros desejados, além de precisar obrigatoriamente dos dados

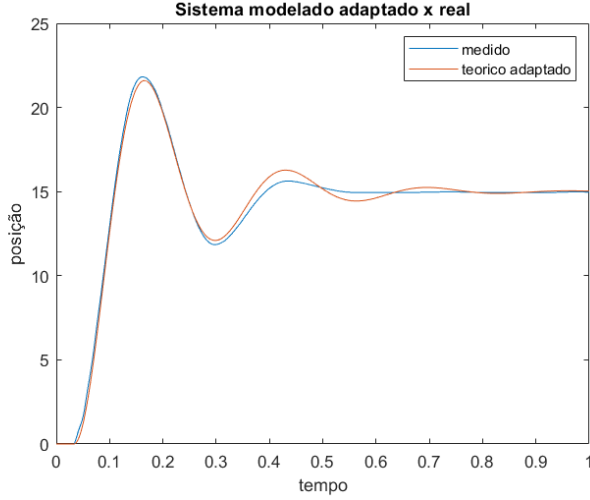


Figura 10: Modelo medido x estimado adaptado [cm x seg]

da tabela inicial. O resultado é obtido e apresentado pela Fig.[11].

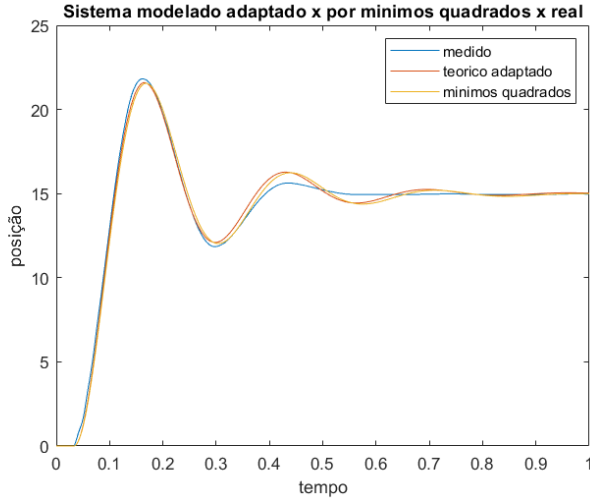


Figura 11: Modelo medido x estimado adaptado x TFEST [cm x seg]

Para melhor comparação, utiliza-se a função `goodnessOfFit` para encontrar o erro médio quadrático das formas de onda estimadas com a amostragem e o erro obtido para o modelo teórico é de 0.0759 enquanto o erro do modelo obtido pela `tfest()` é de 0.0860. Com isso, a estimação pode ser definida como ótima, dado o valor do erro médio quadrático.

3.2.1 Identificação dos parâmetros em Malha Aberta

A partir do desenvolvido da seção 3.2, tem-se uma boa estimação para o sistema em Malha Fechada. Entretanto, para construção do controlador PI é necessário obter seu modelo de malha aberta de acordo a Eq.[36]. De tal modo que como o ganho proporcional utilizado para obtenção de dados foi 420. A equação estimada de malha aberta é:

$$\frac{Y(s)}{R(s)} = \frac{1.429}{s^2 + 12.4s} e^{-0.033s} \quad (39)$$

3.3 Realização

Para se desenvolver um projeto no espaço de estados do servomecanismo é necessário representar o mesmo no espaço de estados. Há inúmeras realizações possíveis porém optou-se pela mais imediata onde as variáveis de estados são a posição e a velocidade definidas pela variável de saída do sistema. Portanto, segue que:

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \quad (40)$$

Seguindo da Eq.[8], realizando a transformada de Laplace inversa, obtêm-se a seguinte equação diferencial:

$$\ddot{y} = br - a\dot{y} \quad (41)$$

Sendo r o valor de referência de entrada, representado geralmente por u no espaço de estados.

Segue algebricamente das Eqs.[40,41], então, a seguinte realização:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ b \end{bmatrix} \mathbf{r} \quad (42)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} \quad (43)$$

Para essa solução, é necessário desconsiderar o efeito do atraso de 0.033 segundos na entrada, já que caso contrário levaria a uma complicação indesejada no projeto ao se adicionar um novo vetor de estados na forma de uma perturbação, o que foge do interesse do projeto. Ademais, o efeito do atraso pode ser desprezado dado seu pequeno valor, e a representação no Matlab é obtida pela função `ss` e apresentada a seguir, pelas Eqs.[44,45].

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -12.4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1.429 \end{bmatrix} \mathbf{r} \quad (44)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} \quad (45)$$

Pode-se avaliar a controlabilidade e a observabilidade do sistema pelas funções `ctrb` e `obsv` do Matlab, que calculam respectivamente os resultados das Eqs.[20, 19], sendo que elas retornam a determinante das respectivas matrizes de controlabilidade, cujo valor é de $3.60 \cdot 10^5$, e observabilidade, cujo valor é de 1. Como ambos os valores diferem de zero, o sistema sob essa realização é tanto observável quanto controlável, o que é de interesse para o projeto do controlador PI via espaço de estados.

3.4 Discretização

Como o controlador em questão será digital, se faz necessária a discretização do modelo no espaço de estados para os cálculos posteriores. A função `c2d` do Matlab é capaz de realizar a transformação do sistema no domínio contínuo para discreto, com padrão de discretização a ser o segurador de ordem zero, ao determinar o período de amostragem. Na coleta de dados para identificação, o período de amostragem foi de 1 milissegundo, contudo como outros sistemas podem ser mais lentos, de modo que foi adotado o período de amostragem de 11ms, que é igual a um terço do atraso percebido no sistema.

Avaliando o número de amostras por ciclo amortecido do sistema, pela Eq.[14], percebe-se que para o referido valor de período de amostragem, há aproximadamente 24 amostras por ciclo de oscilação, o que confere um bom número de amostras ao sistema,

porém, não é um valor muito elevado de modo a possivelmente exigir muito do sistema.

Vale ressaltar que como foi feita uma simplificação nas equações de espaço de estados contínuo para não se adicionar o atraso, a Eq.[39] foi discretizada com a função `c2d`. Com isso, obtêm-se:

$$\frac{Y(s)}{R(s)} = z^{-3} * \frac{8.266e - 05z + 7.899e - 05}{z^2 - 1.873z + 0.8725} \quad (46)$$

Uma vez com o sistema discretizado, a função `canon`, que utiliza a decomposição Schur e conversão para espaço de estados, transforma a Eq.[46] para a forma canônica observável. O resultado é dado pelas Eqs.[47,48]

$$\mathbf{x}[k+1] = \begin{bmatrix} 0 & -0.8725 \\ 1 & 1.873 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \mathbf{r} \quad (47)$$

$$\mathbf{y}[k] = [8.266e - 05 \quad 0.0002338] \mathbf{x}[k] \quad (48)$$

3.5 Características do Sistema

Para a etapa de determinação dos polos do sistema a serem implementados pela realimentação de estados, pode-se utilizar duas das três equações Eqs.[9, 10, 11], já que há apenas duas variáveis no sistema de segunda ordem.

Portanto, para definir os polos desejados no sistema, foi pensado no exemplo real de um carrinho seguidor de linha. Esse carrinho quando está a procura da linha, seja por ter realizado uma curva ou se perdido, anda no sentido que pensa estar a linha. Entretanto, ele não consegue prever onde a linha estará, portanto quando é encontrada, o carrinho passa diretamente por ela e com isso ele oscila em torno da linha até estabilizar nela novamente. Assim, será utilizado os parâmetros de tempo de acomodação e máximo sobressinal, a ser definidos pelos valores:

$$t_s = 1s \quad M_p = 0.25 \quad (49)$$

Um código no Matlab foi desenvolvido para facilitar o cálculo da posição dos pólos desejada com base nesses parâmetros, e os polos são encontrados utilizando da Eq.[12]. O código do matlab também consegue encontrar os pólos caso a característica desejada

seja o tempo de pico, basta escolher duas das 3 características. Porém para o exemplo dado, Máximo sobressinal e tempo de acomodação cumprem melhor o papel.

Por fim, os pólos definidos para o sistema em malha fechada, que caracterizam o comportamento transitório do sistema, são:

$$polos = 0.9522 \pm j0.0953 \quad (50)$$

Quanto ao comportamento estacionário do sistema, o erro se relaciona apenas à possível imprecisão da leitura de posição e atuação, tendo em vista que, pelo uso de integradores para o controlador PI, o sistema possui erro nulo, o que é adequado para esse tipo de sistema.

3.6 Observador

O interesse é que o observador responda mais rápido que o sistema para que compense a diferença de erro na estimação pela diferença entre o estado $\tilde{\mathbf{K}}$ e \mathbf{x} , o que significaria que os polos do observador devem ficar mais próximos da origem. Porém, quanto mais rápido, mais sensível se torna o sistema, o que pode ser um problema para imprecisões e ruído.

A princípio é definido que os polos do observador serão 0.1 o tamanho dos polos do sistema desejado. Utiliza-se a função `place` do Matlab, que recebe como argumento as matrizes \mathbf{G} e \mathbf{H} em espaço de estados no caso discreto, contudo como procuramos a matriz do observador foram utilizadas as matrizes \mathbf{G}^T e \mathbf{C}^T . Assim, pode-se encontrar a matriz de ganho do observador, \mathbf{K}_e , dada pela Eq.[51], que é a transposta da matriz que `place` retorna, neste caso.

$$\mathbf{K}_e = \begin{bmatrix} -3,2959e+03 \\ 8,3604e+03 \end{bmatrix} \quad (51)$$

3.7 Cálculo das matrizes \mathbf{K}_1 e \mathbf{K}_2

Sabendo os polos desejados do sistema de malha fechada, pode-se definir as matrizes \mathbf{K}_1 e \mathbf{K}_2 . Primeiro, são definidas $\hat{\mathbf{G}}$ e $\hat{\mathbf{H}}$ pela Eq.[32]. Em seguida, utilizando a função `place` com os polos e as matrizes, pode-se encontrar $\hat{\mathbf{K}}$.

Vale lembrar que o controlador PI adiciona um estado ao sistema, logo a matriz $\hat{\mathbf{K}}$ possui mais um autovalor, escolhido para ser o mais rápido possível, portanto zero.

Através da Eq.[33], é possível obter facilmente as matrizes, apresentadas no conjunto da Eq.[52].

$$\mathbf{K}_1 = [70.26] \quad \mathbf{K}_2 = [0,9623 \quad 0,9294] \quad (52)$$

3.8 Avaliação Teórica do Sistema

Após os passos descritos nas seções anteriores, todas as matrizes necessárias para a construção do Controlador PI para o sistema foram calculadas, sendo estas: \mathbf{K}_1 , \mathbf{K}_2 e \mathbf{K}_e .

A ser lembrado que é necessário utilizar o observador de estados, visto que o sistema não possibilita o acesso de todos os estados, somente da saída. Assim, um observador conforme a Fig.[7] foi acoplado ao sistema do controlador PI da Fig.[8], cujo objetivo é fornecer a realimentação de estados teórica para o sistema. De tal modo, o controle será tão bom quanto for a identificação do sistema.

Um sistema teórico foi disponibilizado para a execução de testes via Simulink. Com ele, o sistema representado pela Fig.[18], em anexo, foi realizado.

Em que:

- Bloco verde: Sistema teórico disponibilizado
- Retângulo vermelho: Observador de Estados
- Retângulo Azul: Integrador

Foi modelado como o sistema identificado reage ao controle com as matrizes \mathbf{K}_1 e \mathbf{K}_2 , a ser montado conforme a topologia representada pela Fig.[8] e apresentado pela Fig.[19], em anexo.

Em que em ambos modelos, os componentes de ganho dos sistemas são as matrizes, multiplicando no formato $\mathbf{K}\mathbf{u}$, \mathbf{K} a matriz e entrada \mathbf{u} .

Para fins de análise, a Fig.[20], em anexo, apresenta as formas de curva da saída do sistema teórico e identificado quando sob ação do controlador PI proposto. Na imagem estão destacados o valor de pico, o valor em 1s de simulação e o tempo em que o sistema

chega e permanece no valor de referência. Com isso, observa-se:

- $M_p = \frac{18.51-15}{15} = 0.234$;
- Em 1 segundo, o sistema se aloca entre 14.7 e 15.3, com 2% da referência, logo, acomodado;
- O sistema eventualmente foi para o estado estacionário com erro nulo.

De tal maneira que o valor desejado para máximo sobressinal era de 25%, a ter alcançado somente 23,4%, com um erro de 6.4%. Percebe-se também que o sistema já está acomodado quando chega ao tempo esperado de 1s. Assim, o sistema teórico é declarado como de bom desempenho com o controlador PI e o observador de estados.

3.8.1 Resposta à perturbação

Para confirmar a robustez do projeto de controlador, testes com perturbação do tipo degrau foram realizados. Os testes foram executados adicionando a entrada degrau como perturbação no sinal de controle enviado à planta, cujo diagrama é representado pela Fig.[21] e também no sinal de saída do sistema, representado pela Fig.[22], em anexo.

Para o sistema com perturbação na ação de controle, valores de ganho para a perturbação degrau que variaram de 1 a 5000 foram testados. Estes valores são transformados em Volts pelo ganho do modelo físico. De tal maneira, foi observado que o aumento do ganho da perturbação afetava principalmente o máximo sobressinal do sistema, com a curva permanecendo semelhante à sem perturbação. A partir de 5000, a ação de controle começa a sofrer saturação e apesar do sistema conseguir converter isso para não saturar, a resposta transitória difere muito do esperado. Assim, segue a resposta do sistema a entrada 3000 demonstrado pela Fig.[12].

Nota-se que o sistema ainda reage de maneira semelhante ao esperado, dada a planta identificada, contudo há um adiantamento da resposta causado por mais energia ser passada ao sistema, logo executando o tempo de pico mais rapidamente. Ainda mais, foi inserida a ação de controle nesta imagem,

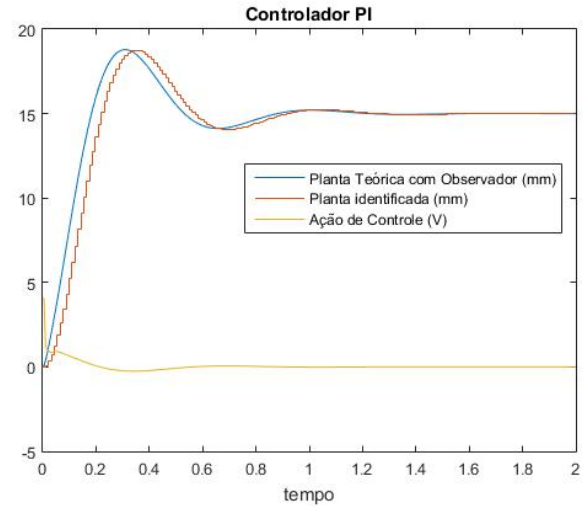


Figura 12: Resposta com perturbação na ação de controle [mm x seg] e [V x seg] para a ação de controle

percebe-se que ela tem um pico inicial, mas conforme o sistema chega ao estado estacionário, o sinal tende a zero, como esperado.

Para a perturbação na saída, como ela afeta tanto a realimentação de saída como o cálculo realizado pelo observador, o teste foi feito somente com perturbação no observador, dessa vez com entradas em metros, que é unidade de saída controlada. Diferentemente do caso anterior, este teste abrangeu somente valores de ganho para a perturbação degrau de 1 a 20, pois logo a ação de controle torna-se volátil inicialmente e apesar do sistema não ficar instável, há grande mudança em sua resposta transitória. Foram obtidos bons resultados até o ganho 5 de perturbação, demonstrados pela Fig.[13].

O valor de saturação proposto para a ação de controle foi ± 8 . Com isso, nota-se que a ação de controle chegou aos pontos de saturação por conta da perturbação e ainda assim levou o sistema ao estado estacionário. Além disso, o sistema se tornou de fase não-mínima, já que começa indo para a direção contrária a devida.

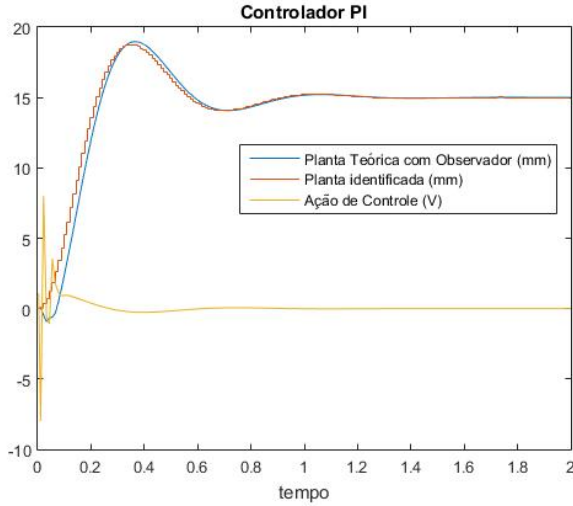


Figura 13: Resposta com perturbação na soma das saídas do observador [mm x seg] e [V x seg] para a ação de controle

3.8.2 Resposta a outras entradas

A continuar a busca quanto a robustez do projeto, outros sinais de entrada foram fornecidos ao sistema para análise da eficiência. Com isso, dois testes que fazem sentido para a problemática são trabalhados: variação da amplitude do degrau, que indica uma mudança no ângulo da curva para o carro segue linha, e excitação do sistema com uma onda quadrada de amplitude 15 e 0.3Hz de frequência, para verificar o quão bem o projeto faria curvas em sequência pouco depois de chegar no tempo de acomodação. A frequência da onda foi escolhida de maneira a ter folga no tempo de acomodação do sistema, assim fica mais fácil identificar como ele reage a nova excitação.

Assim, primeiramente, o teste com outras amplitudes para a entrada degrau foram realizadas com valores de amplitude de 10 e 20. Sendo apresentadas as respostas que o sistema obteve respectivamente pelas Figs.[14, 15].

Em ambos os casos, a resposta do sistema às novas entradas foram muito boas, seguindo quase que perfeitamente o sistema identificado, com a maior

mudança no máximo sobressinal, entretanto tal diferença não chegou a ser maior que 2% em ambos os testes.

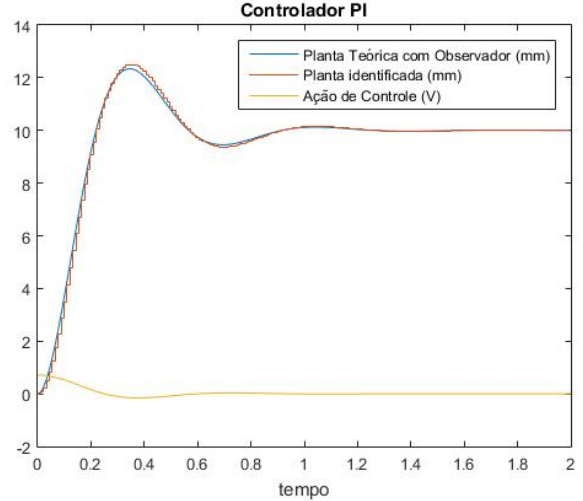


Figura 14: Resposta com Entrada Degrau de Amplitude 10 [mm x seg] e [V x seg] para a ação de controle

Em sequência, foi testado como o sistema reage a um sinal de entrada no formato de uma onda quadrada de amplitude 15 e frequência 0.3 Hz. A resposta que o sistema obteve é dada pela Fig.[16].

Com este resultado, nota-se que o sistema teórico ainda assim segue o formato do sistema identificado. Contudo, apesar do sistema ainda ter tempo de acomodação de aproximadamente de 1 segundo, como ele ainda oscila um pouco antes de chegar no valor de estado estacionário, é possível que o sistema não consiga acompanhar várias curvas em sequência e se perca, caso a frequência das curvas seja maior que 0.5 Hz. Visto que segundo a Fig.[20], o sistema só chega no espaço estacionário em 1.9 segundos, que equivaleria a 0.52 Hz.

3.8.3 Mudança de Parâmetros da Planta

Para aprofundar os testes de robustez da planta, mudança de alguns parâmetros da planta disponibilizada foram feitas. Dado os fatores da planta e o

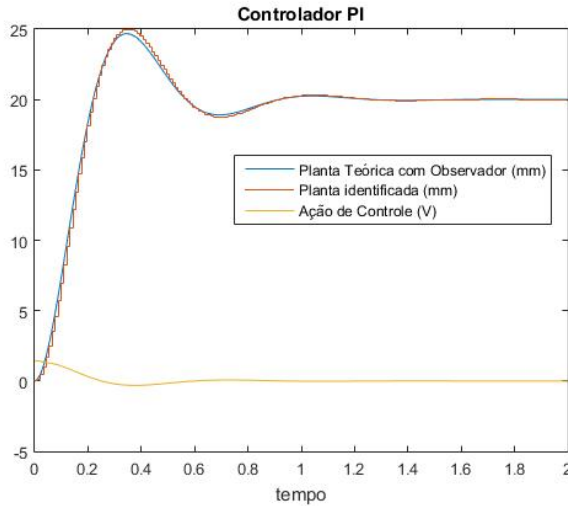


Figura 15: Resposta com Entrada Degrau de Amplitude 20 [mm x seg] e [V x seg] para a ação de controle

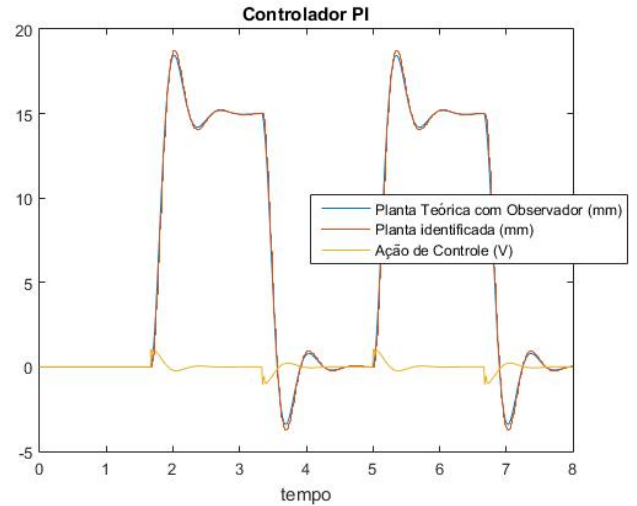


Figura 16: Resposta com Entrada Onda Quadrada [mm x seg] e [V x seg] para a ação de controle

sistema proposto, pensou-se em alterar dois valores: amortecimento viscoso e força eletromotriz. Isso se deve a ambos poderem sofrer alterações naturais com o tempo.

Assim sendo, a mudança dos parâmetros ocorreu de maneira a triplicar o valor da força eletromotriz e reduzir na metade o amortecimento viscoso, dado o desgaste do carro. Com isso, o sistema montado apresentou a seguinte resposta, ainda com o mesmo controlador, conforme a Fig.[17].

Dado que o sistema um observador de estados, que faz com que o sistema tenda ao identificado, não há grande prejuízo no controle do sistema. Entretanto, ainda é observável que todas as três características do sistema tiveram um aumento em seus respectivos valores. A mudança dos parâmetros indica um desgaste e maior esforço do sistema para chegar ao ponto esperado, o que justificaria essa mudança.

3.9 Avaliação prática do sistema

Para testar o controlador desenvolvido na bancada prática disponibilizada, é necessário o uso do arquivo Simulink .mdl, diferente do atualmente usual .slx.

Como com o caso teórico, também foi disponibilizado o bloco Quarc que controla o sistema, de tal maneira que bastava somente inserir os blocos do controlador, que não são parte da planta, junto ao sistema Quarc para realizar o teste.

Entretanto, a verificação da teoria não ocorreu bem como previsto. Há uma diferença entre versões de software Matlab usado para o experimento em relação à versão da bancada, o que impediu o uso do arquivo .mdl gerado a princípio. Assim, o arquivo de simulação precisou ser refeito com blocos adaptados na bancada do projeto. Ademais, o sistema teórico também deixou de apresentar as boas curvas montadas anteriormente, causadas por má configuração dos parâmetros que adaptam o modelo teórico ao físico, da bancada. De tal modo que por questões de tempo, organização e principalmente afinidade com o equipamento, não foi possível montar e testar o controlador em sua totalidade.

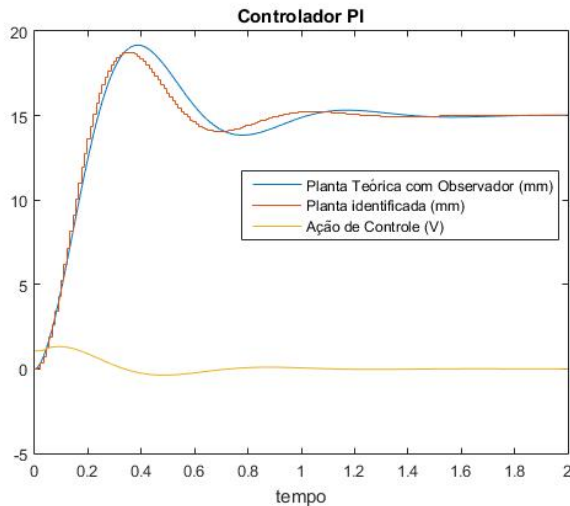


Figura 17: Resposta com mudança de parâmetros [mm x seg] e [V x seg] para a ação de controle

4 Conclusão

Desenvolver um controlador PI para um sistema com servomotor linear teve como principal alvo concluir e fixar os conhecimentos na área de controle digital. De tal maneira a ser possível aplicar e re-visitar conceitos vistos como o teorema de Nyquist, espaço de estados via controle discreto, projeto e implementação física de sistemas, além da modelagem do mesmo. Ademais, permitir a formulação prática de um sistema real permitiu o desenvolvimento de habilidades em identificação de sistemas e metodologias para tratar de problemas ao longo do projeto.

A totalidade do trabalho demonstra o funcionamento da teoria de controle. Primeiramente, o trabalho consistiu na modelagem e identificação do sistema físico que estava na bancada, em que foi obtido um erro quadrático de 0.076. Em seguida, houve a aplicação direta de conhecimentos para a definição das matrizes de controle por realimentação de estados utilizadas \mathbf{K}_1 e \mathbf{K}_2 . Também houve a necessidade do uso de um observador para resolução do problema, gerando a matriz \mathbf{K}_e . Por fim, em posse de todos estes elementos, foi possível verificar o compor-

tamento dinâmico e estacionário adequados, definidos com parâmetros de projeto de máximo sobressinal e tempo de acomodação de 2%.

Apesar dos testes práticos não terem funcionado, é perceptível que a aplicação correta da teoria resulta em bons resultados. Identificou-se que a falha prática não se deve a um erro na teoria e sim a um montagem do sistema para testes. De tal maneira, o controlador projetado neste texto é válido e pode ser desenvolvido e utilizado posteriormente, com as devidas adaptações no ambiente de testes, para se obter sucesso também no aspecto prático e possivelmente validar os aspectos de resposta com as imperfeições desconsideradas dado o escopo de aprendizado aqui adotado. Pode-se buscar por uma modelagem que vise um estudo da viabilidade de uma maior robustez à perturbações gerais, de modo a não permitir a saturação. Uma possível outra melhoria envolveria a implementação de um filtro de Kalman, de modo a filtrar demais ruídos.

5 Referências

- [1] OGATA, K. **Discrete time control systems**. 2nd ed., Prentice-Hall, 1995.
- [2] OGATA, K. **Engenharia de Controle Moderno**. 5ª ed., Pearson, 2011.
- [3] **LINEAR SERVO CONTROL LAB - A Modular Lab for Teaching of Controls, Robotics and Mechatronics.**, Plataforma virtual Quanser
- [4] FERREIRA, H. **Notas de aula da disciplina de Controle Digital**, Plataforma virtual Aprender3

6 Anexos

6.1 Imagens grandes

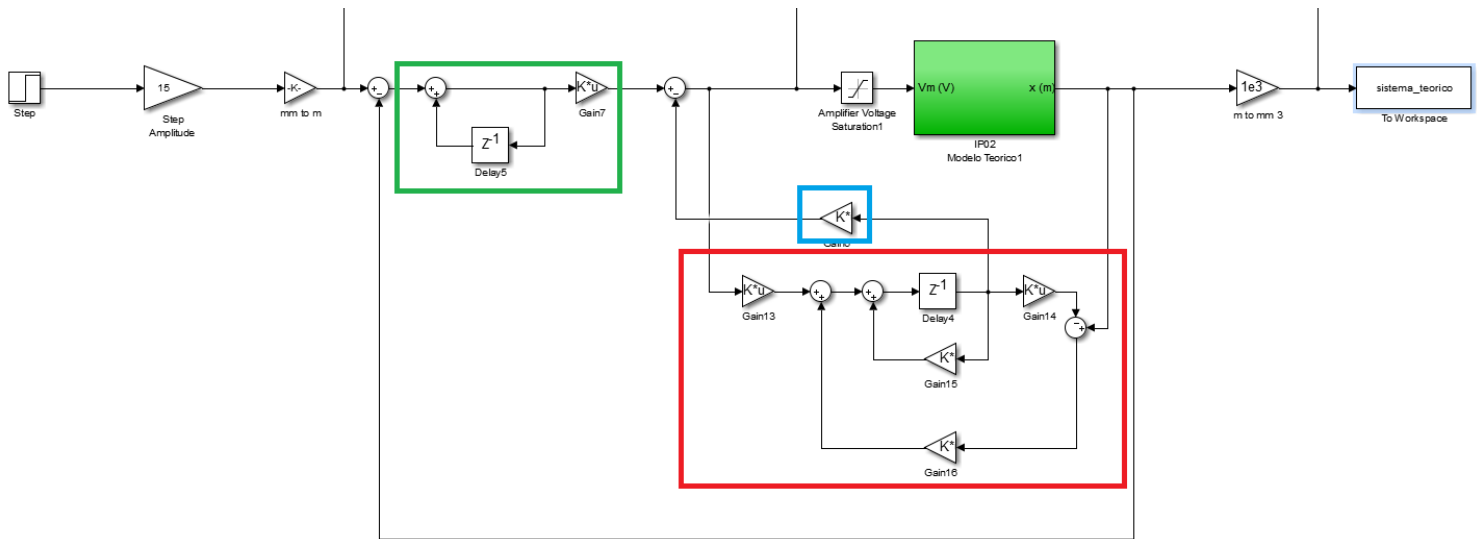


Figura 18: Modelagem Teórica no Simulink

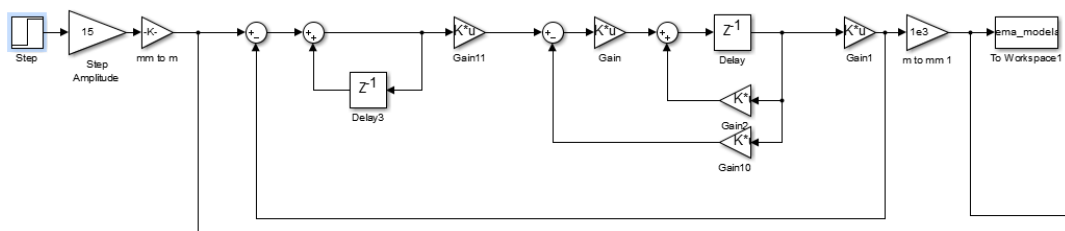


Figura 19: Modelo identificado no Simulink

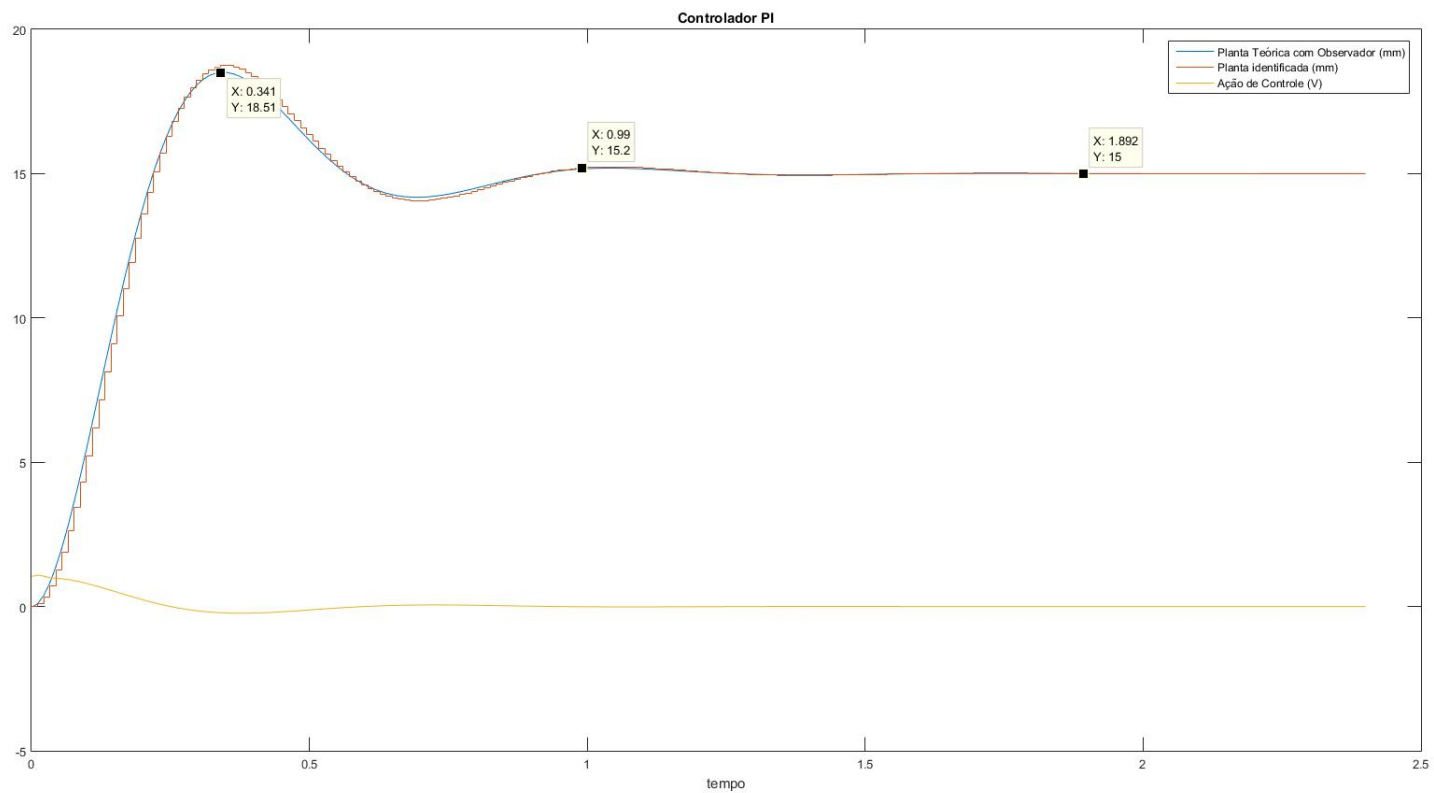


Figura 20: Resposta degrau no sistema controlado [cm x seg] e [cm x seg] para a ação de controle

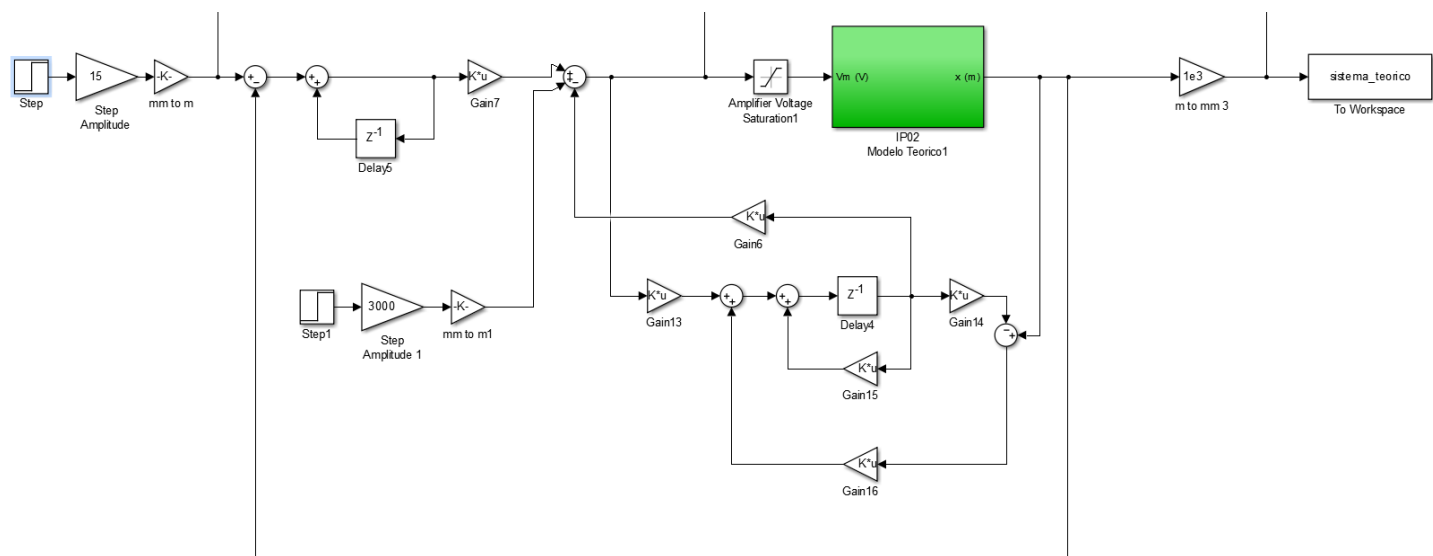


Figura 21: Modelo teórico com Perturbação na Entrada

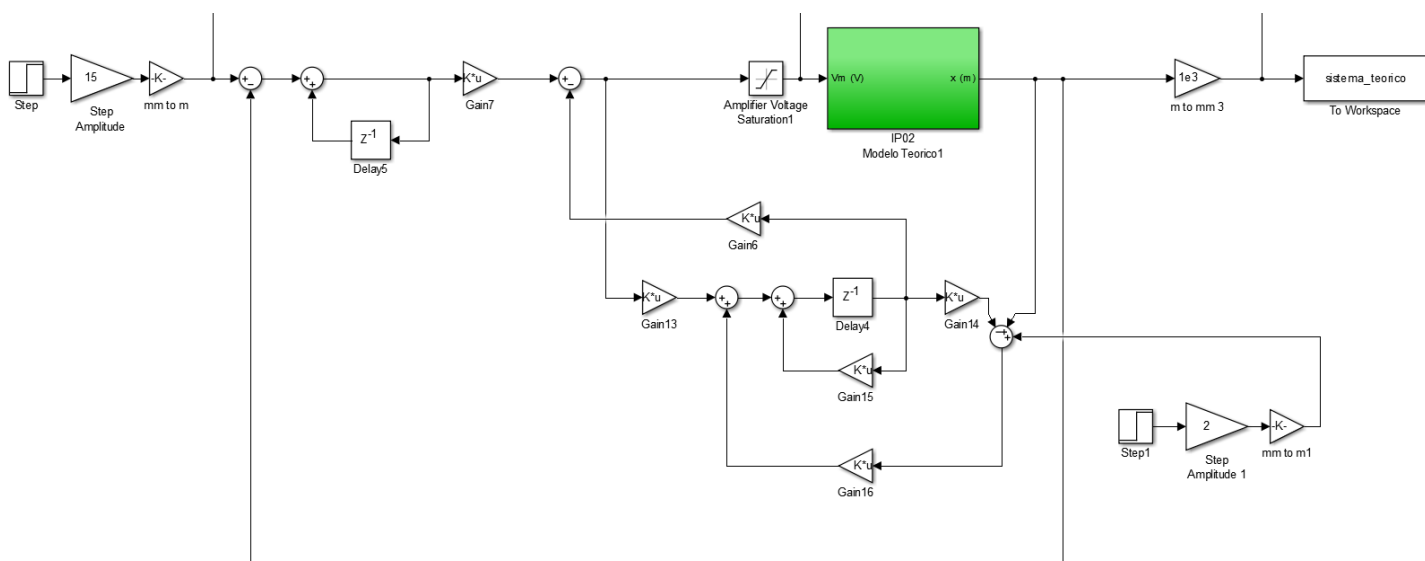


Figura 22: Modelo teórico com Perturbação na soma das saídas para do observador

6.2 Código Matlab

```
1 % Universidade de Brasilia - Faculdade de Tecnologia
2 % Laboratorio Controle Digital - semestre 2021/2
3 % Alunos:
4 %     Gabriel Tambara Rabelo - 18/0017021
5 %     Pedro Pereira Nunes - 17/0020738
6
7 % tendo obtido a matriz de dados x da simulacao:
8
9 % separando vetor temporal das medicoes
10 time = x(:,1);
11
12 % separando vetor sinal de saida das medicoes
13 output = x(:,2);
14
15 % separando vetor de input das medicoes
16 input = x(:,3);
17 %%
18
19 % para corrigir os vetores tempo e saida para iniciarem com t=0:
20 %
21 aux = [0:0.001:0.032];
22 time = [aux time.'];
23 time = time.';
24 output = [zeros([1 33]) output.'];
25 output = output.';
26 aux = 15+zeros(1, 33);
27 input = [aux input.'];
28 input = input.';
29
30 % buscando o valor maximo da saida
31 saida_max = max(output);
32
33 % calculado o maximo sobressinal dada uma entrada de 15
34 maxss = (max(output)-15)/15;
35
36 % buscando tempo de pico
37 tp = time(find(output==saida_max, 1));
38
39 % calculando o fator de amortecimento
40 xi = -log(maxss)/(((pi^2)+((log(maxss))^2))^(0.5));
41
42 % calculando a frequencia amortecida
43 wd = pi/tp;
44
45 % calculando frequencia natural
46 wn = wd/((1-xi)^(1/2));
47
48 % definindo a variavel s de frequencia complexa
49 s = tf('s');
50
51 % modelando o formato geral para um sistema de segunda ordem se o erro
52 %sistema = (wn^2)/(s^2 + 2*s*xi*wn + wn^2);
53
54 % como o vetor de dados inicia em 0.033 segundos, percebe-se que o modelo
55 % tem um atraso, o seu sistema representando esse atraso seria na forma:
```

```

56 %
57 %  alternativa para sistema com vetores de entrada corrigidos (iniciados a
58 %  partir de t=0):
59 %
60 sistema = ((wn^2)*exp(-s*0.033))/(s^2 + 2*s*xi*wn + wn^2);
61
62 %  definindo a entrada degrau para o vetor temporal
63 %t = 0.033:0.001:1
64 u = zeros(length(time),1);
65 u(time>=0) = 15;
66
67 %  declarando saida do sistema
68 output_teorico = lsim(sistema,u,time);
69
70 %  plotando o resultado
71 figure
72 plot(time, output, time, output_teorico)
73 hold on
74 title('Sistema modelado x real');
75 legend('medido','teorico');
76 xlabel('tempo');
77 ylabel('posicao');
78
79 %  fazendo adaptacoes heurísticas para melhorar a modelagem, usa-se:
80 wn_adpt = 24.5;
81 xi_adpt = 0.253;
82 %sistema_adpt = (wn_adpt^2)/(s^2 + 2*s*xi_adpt*wn_adpt + wn_adpt^2);
83 %  alternativa para sistema com vetores de entrada corrigidos (iniciados a
84 %  partir de t=0)
85
86 sistema_adpt = ((wn_adpt^2)*exp(-s*0.033))/(s^2 + 2*s*xi_adpt*wn_adpt + wn_adpt^2);
87 output_teorico_adpt = lsim(sistema_adpt,u,time);
88
89 %  plotando o resultado
90 figure
91 plot(time, output, time, output_teorico_adpt)
92 hold on
93 title('Sistema modelado adaptado x real');
94 legend('medido','teorico adaptado');
95 xlabel('tempo');
96 ylabel('posicao');
97
98 %  declarando sistema com modelagem obtida por fitting via metodo dos minimos
99 %  quadrados
100 T = 0.001;
101 np = 2;
102 nz = 0;
103 data = iddata(output, input, T);
104 iodelay = 0.033; %atraso conhecido do sistema
105 tfmmq = tfest(data,np,nz,iodelay);
106 output_mmq = lsim(tfmmq,u,time);
107
108 %  plotando para comparacao das 3 curvas
109 figure
110 plot(time, output, time, output_teorico_adpt, time, output_mmq)
111 hold on
112 title('Sistema modelado adaptado x por minimos quadrados x real');

```

```

113 legend('medido','teorico adaptado','minimos quadrados');
114 xlabel('tempo');
115 ylabel('posicao');
116
117 % avaliacao de qual metodo possui menor erro em relacao a curva medida
118 fitmmq = goodnessOfFit(output_mmq,output,'NRMSE');
119 fit = goodnessOfFit(output_teorico_adpt,output,'NRMSE');
120
121 % realizacao canonica controlavel de sistema_adpt
122 T = T*11;
123 Ganho = 420;
124 b = (wn_adpt^2)/Ganho;
125
126 % definindo realizacao com estados x e x':
127 A = [0 1; 0 -2*xi_adpt*wn_adpt];
128 B = [0 ;b];
129 C = [1 0];
130 D = 0;
131
132 % sistema_adpt_base = ss(A, B, C, D);
133 sistema_adpt_base = (b*exp(-s*0.033))/(s^2 + 2*s*xi_adpt*wn_adpt);
134 sistema_adpt_d1 = c2d(sistema_adpt_base, T,'zoh');
135 sistema_adpt_d = canon(sistema_adpt_d1, 'companion');
136 %sistema_adpt_d = canon(sistema_adpt_d, 'companion'); %retorna na forma canonica
    observ vel
137
138 G = sistema_adpt_d.A;
139 H = sistema_adpt_d.B;
140 C = sistema_adpt_d.C;
141 D = sistema_adpt_d.D;
142
143 % checar controlabilidade (verdadeira se diferente de 0)
144 dc = det(ctrb(sistema_adpt_d));
145
146 % checar observabilidade (verdadeira se diferente de 0)
147 do = det(observ(sistema_adpt_d));
148
149 %DETERMINAR OS POLOS DESEJADOS COM BASE NAS CARACTERISTICAS DO SISTEMA)
150 Ts = 1; %Settling Time;
151 Tp = 0.2; %Peak Time;
152 Mp = 0.25; %Sobressinal;
153 i = 3; %Seleciona qual dois par metros serao utilizados
154 if i==1 %Ts e Tp
155     sigma = 4/Ts;
156     omega_d = pi/Tp;
157 elseif i==2 %Mp e Tp
158     a = -(pi^2)-(log(Mp))^2;
159     c = (log(Mp))^2;
160     zeta = abs(roots([a 0 c]));
161     omega_d = pi/Tp;
162     omega_n = omega_d/(sqrt(1-zeta(1)^2));
163     sigma = zeta(1)*omega_n;
164 else %Ts e Mp
165     a = -(pi^2)-(log(Mp))^2;
166     c = (log(Mp))^2;
167     zeta = roots([a 0 c]);
168     sigma = 4/Ts;

```



```

169     omega_n = sigma/zeta(1);
170     omega_d = omega_n*sqrt(1-zeta(1)^2);
171 end
172
173 r = exp(-sigma*T);
174 teta = omega_d*T;
175 polo1 = r*cos(teta)+r*sin(teta)*j;
176 polo2 = r*cos(teta)-r*sin(teta)*j;
177
178 %POLOS E MATRIZ DO OBSERVADOR
179 V = 0.1; %Quao mais rapido sera o polo do observador
180 r_obs = V*r;
181 polo1_obs = r_obs*cos(teta)+r_obs*sin(teta)*j;
182 polo2_obs = r_obs*cos(teta)-r_obs*sin(teta)*j;
183
184 polos_obs = [polo1_obs polo2_obs];
185 L = place(G', C', polos_obs);
186 L = L';
187
188 %DADOS OS POLOS, CALCULAR MATRIZES K1 e K2
189 %PARA CONTROLADOR PI
190
191 G_chapeu = [G H; 0 0 0];
192 H_chapeu = [0; 0; 1];
193 polos = [polo1 polo2 0];
194 K_chapeu = place(G_chapeu, H_chapeu, polos);
195 I = eye(2);
196 M_aux = [G-I H; C*G C*H];
197 M_I = [0 0 1];
198 K = (K_chapeu+M_I)*inv(M_aux);
199 K1 = K(3);
200 K2 = [K(1) K(2)];
201
202 GCon = G;
203 VMAX_AMP = 8;
204
205 sim('Teorico_plus_estimado');
206
207 % plotando o resultado
208 figure
209 plot(sistema_teorico);
210 hold on
211 plot(sistema_modelado);
212 hold on
213 plot(acao_controle);
214 title('Controlador PI');
215 legend('Planta Teorica com Observador (mm)', 'Planta identificada (mm)', 'Acao de Controle (V)');
216 xlabel('tempo');
217
218 % As caracteristicas testadas tem que ser manualmente mudadas no arquivo .slx

```